Proof of Coherence

A Governance & Verification Layer for Federated AI Meshes

Sylvain Cormier
Paraxiom Research
research@paraxiom.io

January 2026

Problem Statement

Federated and distributed AI systems rely on untrusted, heterogeneous nodes contributing model updates, inference outputs, or local training work. Today, acceptance criteria are weak: statistical aggregation, trust in operators, or coarse anomaly detection.

This creates four risks:

1. **Hallucinated or incoherent contributions** accepted into shared models
2. **Silent destabilization** of global models from rogue nodes
3. **No objective notion of "useful work"** in AI meshes
4. **Wasted energy** on redundant verification through recomputation

There is currently no equivalent of "proof of useful work" for distributed AI.

Core Idea

Proof of Coherence is a lightweight verification layer that proves an AI contribution was internally coherent, non-divergent, and safe to accept into a shared AI system—**without trusting the node** that produced it and **without re-running the model**.

> *We do not prove that an AI is right—we prove that it stayed coherent.*

What "Coherence" Means (Operationally)

Each AI contribution produces, alongside its output, a **coherence fingerprint**:

Signal	What It Measures	Computation
Attention entropy	Distribution stability across heads	O(n) per layer
Spectral CV	Coefficient of variation of eigenvalues	$O(n^2)$ once
Topological consistency	Deviation from expected neighborhoods	O(n) per layer
Drift rate	Change in hidden state norms	O(1) per layer

These signals are **architecture-aware** (calibrated per model family), **cheap to compute** (∼5% overhead), and **deterministic to verify**.

Integration into Federated Learning

Existing flow: Node → Local Training/Inference → Update Sent → Aggregation

With Proof of Coherence: Node → Local Training/Inference → Coherence Fingerprint (5% overhead) → Verification Gate (0.1% overhead) → Accepted/Rejected → Aggregation

This is a **sidecar governance layer**, not a rewrite. No changes to training algorithms, model architectures, or orchestration tooling.

Why This Matters

1. Architecture-Sensitive

Different models have different stability envelopes. Our research demonstrates this empirically:

Model	Toroidal Constraint Effect
Phi-2 (2.78B)	50% hallucination **reduction**
TinyLlama (1.1B)	180% hallucination **increase**

Same constraint, opposite effect. Universal fixes don't exist. Governance must be architecture-aware.

2. Hallucination-Aware

Hallucinations are treated as **coherence failures**, not just factual errors. A model that drifts outside its coherence envelope is flagged before its output propagates.

3. Adversary-Resistant

Coherence fingerprints are derived from internal model dynamics. Faking coherence is computationally equivalent to doing coherent work.

4. Energy Efficient

Approach	Energy Cost
Full recomputation	100%
Proof of Coherence verification	**¡ 1%**

For large-scale federated AI: **99%+ reduction** in verification energy. This is green AI governance: trust through mathematics, not brute force.

Proof of Useful Work (Reframed)

Traditional systems prove energy burned (PoW) or capital locked (PoS).

Proof of Coherence proves: **This AI work was useful because it preserved or improved system stability.**

Current Status

- Implemented and tested on real LLMs
- Discovered architecture-dependent effects (critical for governance design)

- Published: DOI: 10.5281/zenodo.18267913
- Open source: github.com/Paraxiom/topological-coherence

What This Enables

Capability	Benefit
Governance primitive	Accept/reject AI contributions objectively
Trust without central control	Nodes prove their own coherence
Energy efficiency	Verify without recompute
Scalability	O(1) verification per contribution

One-Sentence Summary

Proof of Coherence is a verification layer that proves distributed AI work was stable and useful—not just computed—before it is trusted by the system, at 99% lower energy cost than recomputation.

Contact: Sylvain Cormier | @ParaxiomAPI | research@paraxiom.io